REMARKS/ARGUMENTS

Amendments were made to the specification to correct errors and to clarify the specification. No new matter has been added by any of the amendments to the specification.

Claims 1-13 are pending in the present application. Claims 1, 4-5, 9-10, and 13 were amended. Reconsideration of the claims is respectfully requested.

I. Examiner Interview

The examiner is thanked for the courtesy of a telephonic interview, in which support for the claim amendments and the art were discussed. It is understood from that interview that the amendments will overcome the outstanding rejection, but that a further search will be performed.

II. Objection to Claims: Claim 1

The examiner has stated that claim 1 was objected to for the use of "to be". In response, the claim has been rewritten to overcome this objection.

III. 35 U.S.C. § 102, Anticipation: Claims 1, 5 and 10

Claims 1, 5 and 10 stand rejected under 35 U.S.C. § 102 as being anticipated by **Hack** et al., System and Method of Implementing a Virtual Data Modification Breakpoint Register, U.S. Patent Publication No. 2003/0217355 (November 20, 2003) (hereinafter "**Hack**"). This rejection is respectfully traversed.

The rejection states:

> With respect to claims 1, 5 and 10, Hack et al. discloses ... wherein when a first portion of said filesystem is found to be corrupt (See paragraph 0013), said set of instructions are connected to: receive information regarding a location of paid first portion and a perceived corruption ... isolate said first portion of said filesystem while leaving other portions of said filesystem available (See paragraph 0030 and 0039), and provide repair for said filesystem (See paragraph 0031).

Office Action dated June 7, 2006, page 3, first paragraph

The claims have been amended to more clearly recite the invention, specifically reciting that the filesystem in which corruption is found is on *at least one mountable device*, that the corrupt portion is on *a first mountable* device, and that portions of the filesystem that are not corrupt remain available to the system. These features are supported by the application on page 1, line 9 through page 2, line 6, which discusses the context to which the invention is directed.

It is noted that the current application defines an *unmounted* filesystem as one that is *hidden from the operating system* (see page 1, lines 14-15), implying that a mounted filesystem is not hidden from, but available to, the operating system. This coincides with the definition given in Webopedia, the online dictionary of computing, which defines *mounting* as: "*mak[ing] a mass storage device available*".

Amended claim 1 recites:

> 1. A computer system, comprising:
> a first processor connected as a server;
> a plurality of client processors connected to communicate with said first processor;
> a filesystem stored on at least one mountable device and connected for access by said first processor and said plurality of client processors; and
> a set of instructions configured to run on said computer system, wherein when a first portion of said filesystem is found to be corrupt, sid first portion being on a first mountable device of said at least one mountable device, said set of instructions are connected to:
> receive information regarding a location of said first portion and a perceived corruption,
> in response to receipt of said information, isolate said first portion of said filesystem while leaving other portions of said filesystem available,
> while said first portion of said filesystem is isolated, provide repair for said first portion of said filesystem; and
> after repair of said first portion of said file system, remove the isolation of said first portion.

Hack does not anticipate amended claim 1 because **Hack** does not show a number of features of this amended claim, both original features and added features. Most significantly, this reference does not show the actual repair of the filesystem. The rejection cites paragraph [0031] of **Hack** as showing the repair of the filesystem. This paragraph states:

> [0031] The invention assumes that corrupted variables or addresses of corrupted variables are known. If so, a software developer or programmer may specify which address or addresses are to be protected and where in a program the protection is to start. For example, suppose variable V-DMBR is the variable whose data is being corrupted, then the software developer may insert an event-tracing-triggering command at a location in the program where the variable V-DMBR is to begin being monitored. Note that it is well known that an offending variable or its address may easily be determined by first running a program through a debugger.

Hack, paragraph [0031]

Paragraph [0031] does not show repairing the filesystem as recited in amended claim 1; instead, this excerpt addresses the placement of a triggering command to facilitate monitoring of the variable. **Hack** is not directed to the repair of corrupt data, but to a means to detect the manner in which the corruption is occurring and to either jump to a software handler (**310**, Figure 3) or provide a dump to a programmer (**510**, Figure 5). No part of **Hack** addresses the actual repair of corrupt information.

Further, **Hack** does not disclose that the corrupt portion of the filesystem is isolated in response to identification of the corruption or that the isolation is removed after the repair. As **Hack** does not address the actual repair of corrupt information, it is not surprising that this reference does not discuss isolating a portion of the filesystem for the purpose of repair.

Given the features of the claimed invention not met by this reference, the rejection of claim 1 under § 102 over **Hack** is overcome. Claims 5 and 10 have been rejected for reasons similar to claim 1; therefore, their rejection is also overcome. Therefore, the rejection of claims 1, 5 and 10 under 35 U.S.C. § 102 has been overcome.

Furthermore, **Hack** does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. **Hack** provides data regarding corrupt information, but this reference does not provide teachings regarding the repair of discovered corruption. Absent the examiner pointing out some teaching or incentive to modify **Hack** to repair the corruption, one of ordinary skill in the art would not be led to modify **Hack** to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify **Hack** in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

IV. 35 U.S.C. § 102, Anticipation: Claims 1-3, 5-8, and 10-13

Claims 1-3, 5-8, and 10-13 stand rejected under 35 U.S.C. § 102 as being anticipated by **Snyder** et al., Method and System for Verifying and Storing Documents During a Program Failure, U.S. Patent No. 6,816,984 (November 9, 2004) (hereinafter "**Snyder**"). This rejection is respectfully traversed.

The rejection states:

> Regarding claim 1, Snyder et al. discloses ... when a first portion of said filesystem is found to be corrupt (See column 6, line 53-60), said set of instructions are connected to: receive information regarding a location of paid first portion and a perceived corruption (See column 6, line 10-17), isolate said first portion of said filesystem while leaving other portions of said filesystem available (See column 9, line 28-35), and provide repair for said filesystem (See column 9, line 28-35).

Office Action dated June 7, 2006, pages 3-4

Amended claim 1 addresses corruption found on a mountable device that in the past would have been unmounted during the repair process. **Snyder** addresses an application crash and is concerned with the need to save any open files that the application is using. In **Snyder**, the concern is with the possibility of corrupt information in system memory, not directly with corruption in the filesystem itself. Although a mountable device and system memory are both storage, these two types of storage do not share the same concerns. **Snyder** does not disclose the isolation of a portion of the filesystem in response to receipt of the corruption information, nor does this reference disclose that the isolation is removed once the repair is completed.

Since **Snyder** does not show all of the features of claim 1, the rejection of claim 1 is overcome. Claims 5 and 10 have been rejected for reasons similar to claim 1; therefore, their rejection is also overcome. The remaining claims in this rejection are dependent on one of claims 1, 5, and 10; the rejection of these claims is also overcome. Additionally, the dependent claims recite other additional combinations of features not suggested by the reference.

For example, claim 2 recites that "*said set of instructions receives said information from a scout process that traverses the filesystem looking for corruption*". The rejection points to the crash handler of **Snyder** as the equivalent of the claimed scout process. However, although **Snyder** discloses that "*the crash handler examines each file for evidence of corruption*", the files so examined are the *open* files that the application has been using. **Snyder** does not disclose a scout process that traverses the filesystem looking for corruption.

Therefore, the rejection of claims 1-3, 5-8, and 10-13 under 35 U.S.C. § 102 has been overcome. Furthermore, **Snyder** does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. **Snyder** addresses a specific situation in which open files need to be saved and does not discuss making repairs to a portion of a filesystem that is already stored. One of ordinary skill in the art would not be led to modify **Snyder** to reach the present invention when the reference is examined as a whole.

V. 35 U.S.C. § 103, Obviousness: Claims 4, 9, and 13

The examiner has rejected claims 4, 9, and 13 under 35 U.S.C. § 103 as being unpatentable over **Snyder** in view of **Draper** et al., Transaction Clash Management in a Disconnectable Computer and Network, U.S. Patent No. 5,878,434 (March 2, 1999) (hereinafter "**Draper**"). This rejection is respectfully traversed.

Claims 4, 9, and 13 are dependent on claims 1, 5, and 10, respectively. These claims contain the same distinctions over **Snyder** as do their independent claims. Specifically, **Snyder** does not disclose the isolation of a portion of the filesystem while the repair takes place, nor does this reference disclose that the isolation is removed once the repair is completed. Further, **Draper** does not make up for this lack in **Snyder**. **Draper** is reconciling separate versions of a file and uses a lock to enforce order on this reconciliation. There is no suggestion in **Draper** that the lock be used during repair of a filesystem. Therefore, the rejection of claims 4, 9, and 13 under 35 U.S.C. § 103 has been overcome.

VI. Conclusion

It is respectfully urged that the subject application is patentable over **Hack**, **Snyder** and **Draper** and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: August 10, 2006

Respectfully submitted,

/Betty Formby/

Betty Formby
Reg. No. 36,536
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Agent for Applicants